Track Reconstruction in BRAHMS

Kristian Harder, Kansas State University

22 May, 2003

Detector Overview

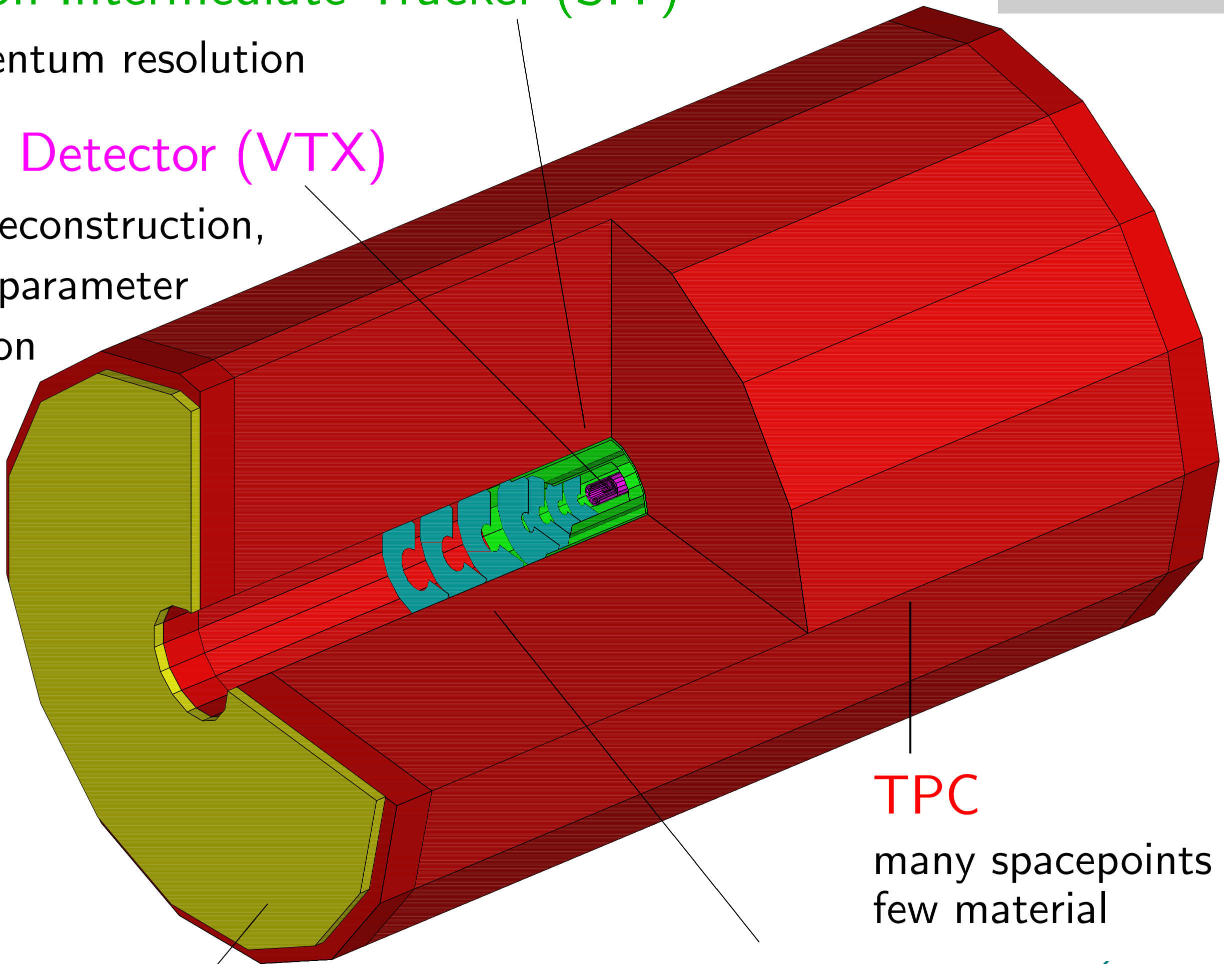
BRAHMS 2.01

Silicon Intermediate Tracker (SIT)

momentum resolution

Vertex Detector (VTX)

vertex reconstruction,
impact parameter
resolution



TPC

many spacepoints
few material

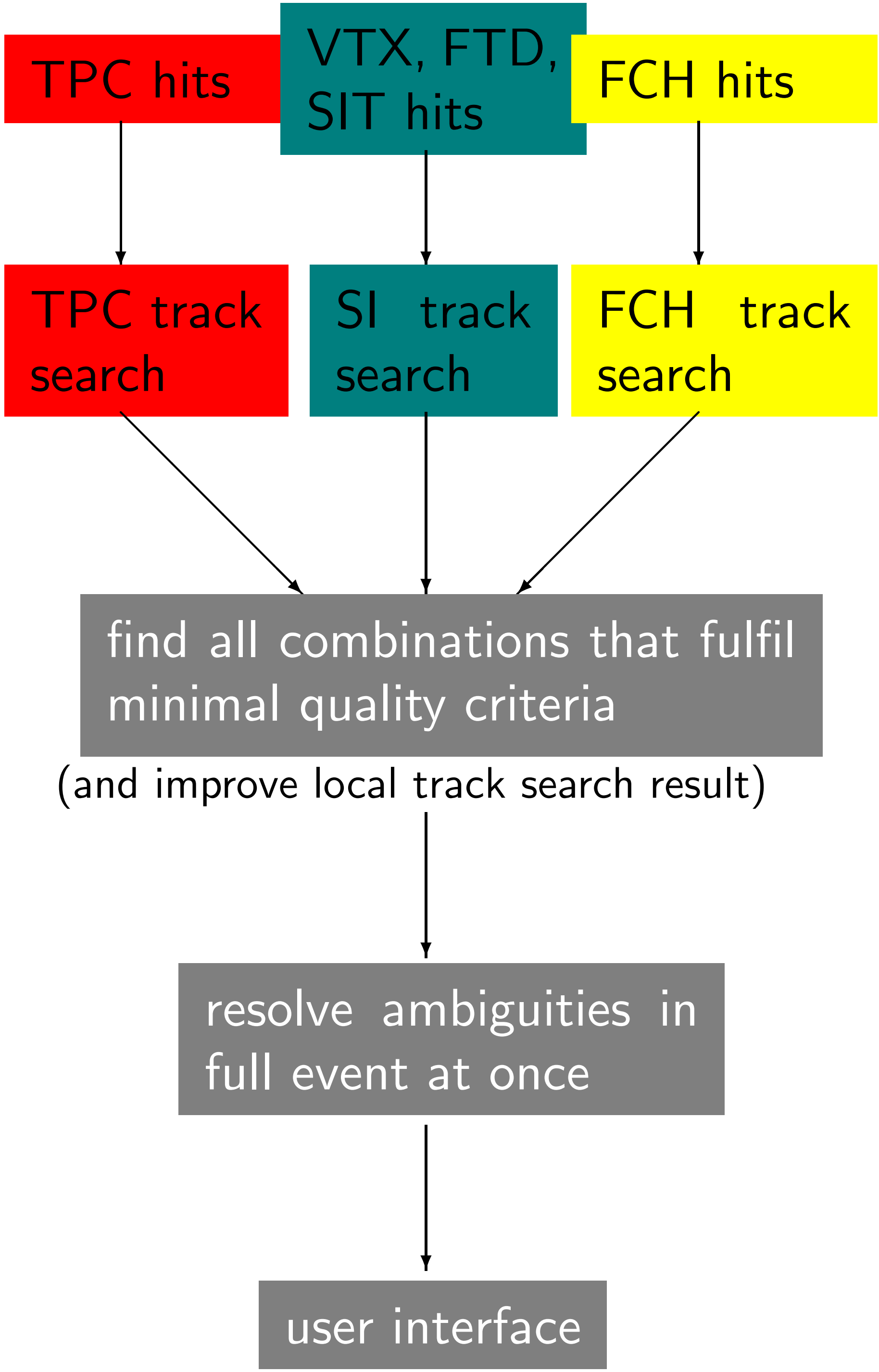
Forward Tracking Disks (FTD)

extend good precision into forward
region (bhabha spectrum!)

Forward Chambers (FCH)

improve TPC in forward direction

Reconstruction Overview



1. local track search with Kalman filters based on software from ALEPH, OPAL
2. connect pieces to ambiguous track candidates using DELPHI software
3. optimize assignment of hits to tracks on global event basis (DELPHI ambiguity resolver)

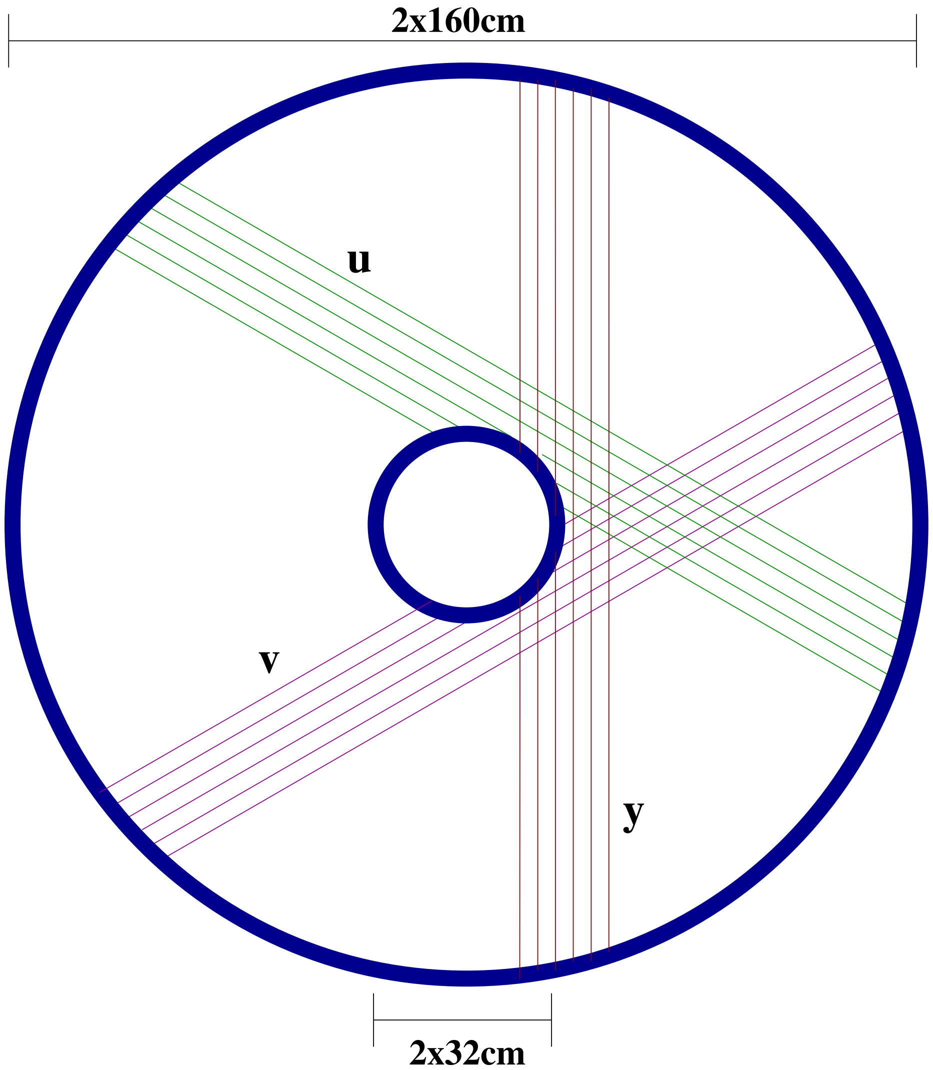
Overview of this talk

- explain algorithms used for FCH
TPC
Silicon Detectors
- concept of the global track search

WARNING: 40,000 lines of code summarized in 30 minutes

This is going to be *very* technical and *very* nasty!

Forward Chambers — FCH



two identical modules behind each TPC endcap

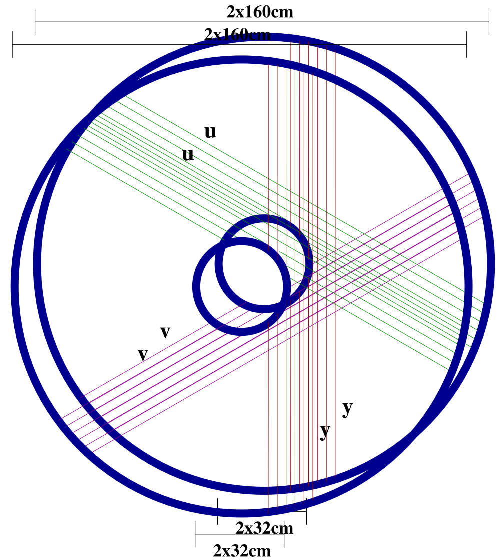
each module has three submodules: x, u, v

each submodule: two staggered wire layers ("double layer")

similar device in DELPHI

$\longrightarrow$ track fit code was available

Forward Chambers — FCH

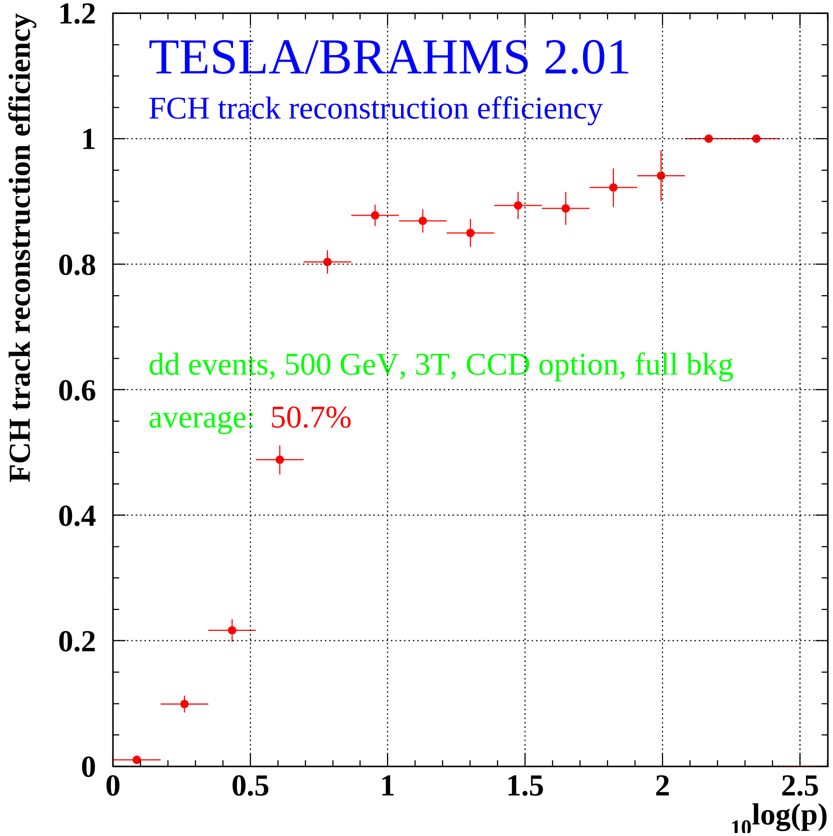


two modules
each with three submodules: x, u, v
each submodule: two staggered wire layers

Track reconstruction algorithm:

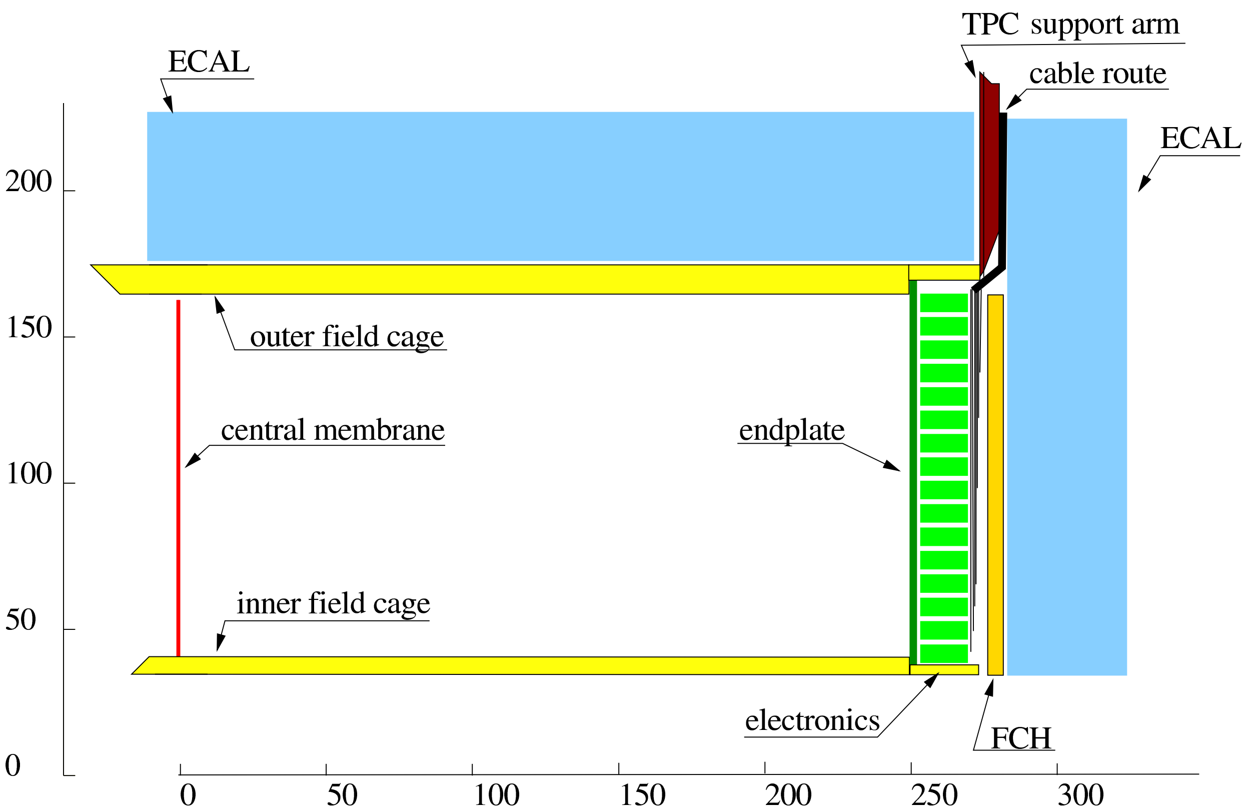
- collect wire hit pairs in double layers compatible with straight track from IP ($\boldsymbol{\Delta r}$)
- combine x, u, v wires to hits (staggered wires not used, inner/outer module separately, combination compatible with straight track from IP)
- select pairs of hits that are compatible with straight track from IP
- do helix fit with all 12 wire layers

Forward Chambers — FCH



FCH standalone track reconstruction efficiency over log(p)
efficiency drops below 10 GeV (straight track requirements)
plot for *all* tracks. muon efficiency is higher

Time Projection Chamber — TPC



200 concentric pad rows

use ALEPH code. additional issue:

track density in high energy jets

$\longrightarrow$ problems due to limited double hit resolution

Time Projection Chamber — TPC

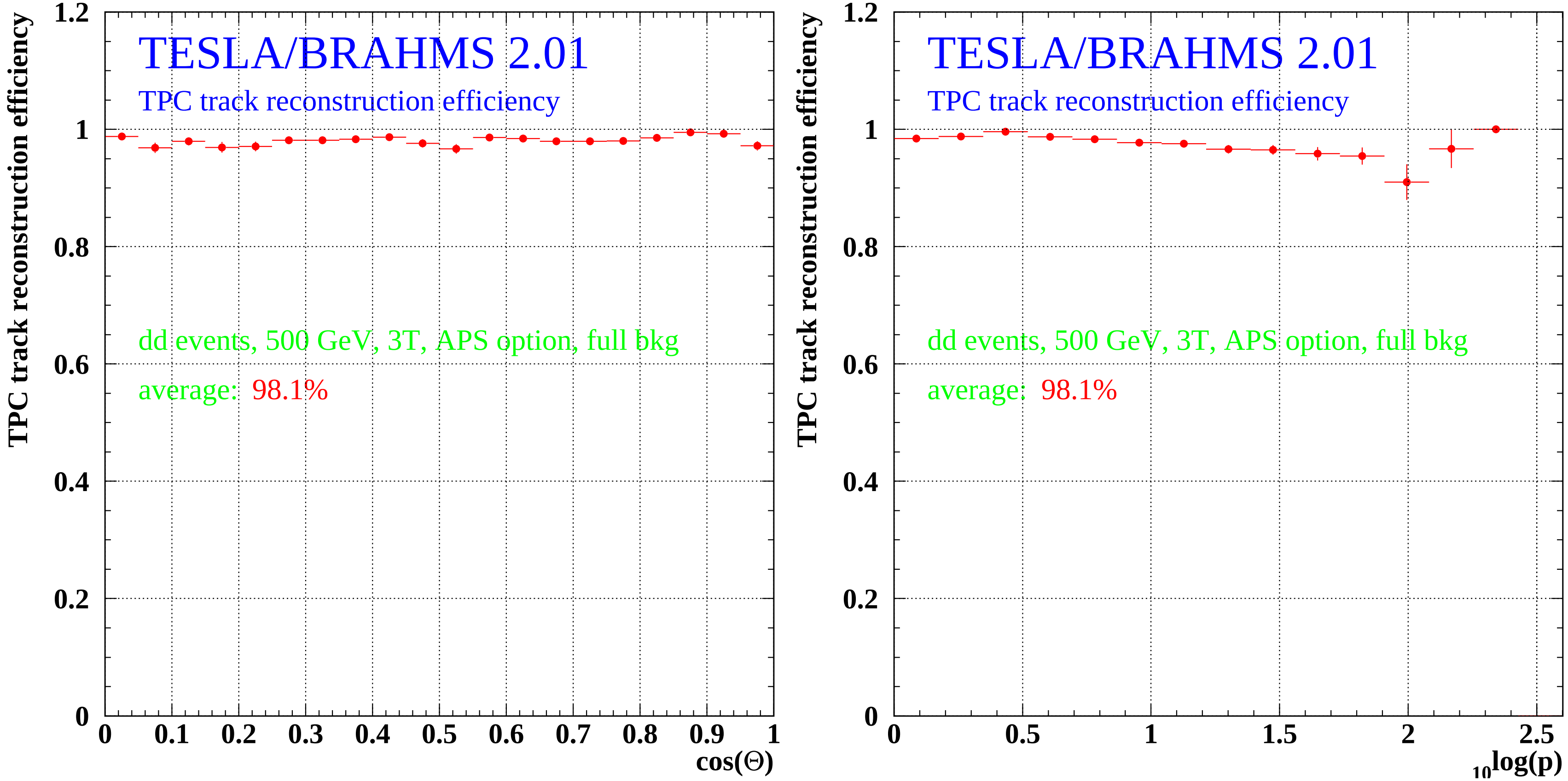
Track Reconstruction Algorithm:

- look for 3 point seeds that give good tracks (consistency; d_0, z_0 cuts)
- take inner 2 hits from above seed; add inner hit to build new seeds; if parameters match previous seed, combine everything to chain
- remove chains that fail helix fit — but split into halves and retry fit
- try to pickup hits on all unused pad rows (take hit with best χ^2)

- check all chains for compatibility (track parameters, pad rows) $\longrightarrow$ e.g. merge branches of spiraling particles
- fit: remove points with bad χ^2, search for kinks
- check all chains for compatibility again
- final refit incl. energy loss

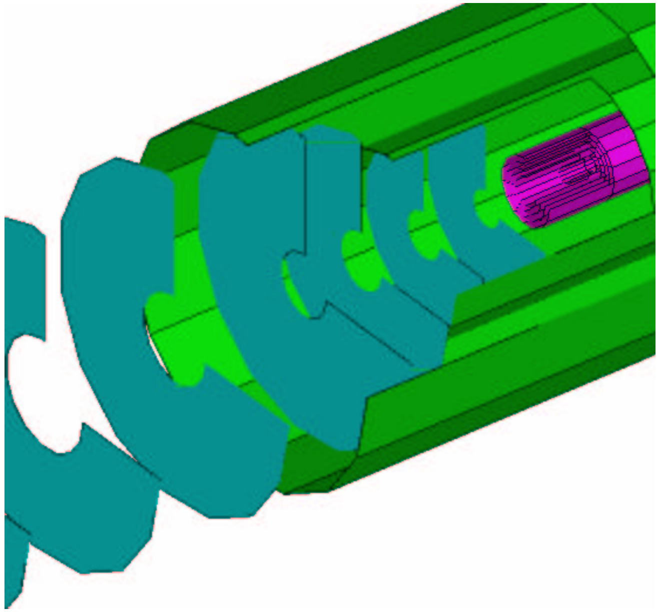
everything above line done *twice* to deal with double hits:
find tracks in outer half only; extrapolate chains to inner half, remove hits in regions where chains overlap, then do full search

Time Projection Chamber — TPC



left: efficiency flat over $\cos(\Theta)$
right: efficiency over $\log(p)$:
  dip at very few very dense tracks (high energy)
  drop below 1 GeV (not shown)

Silicon Detectors — SI



vertex detector (pixels)
forward tracking disks (pixels, strips)
silicon intermediate tracker (strips)

specific problems:

- large background on inner vertex detector layers
- strip detectors $\longrightarrow$ mirror hits
- different detector types (strips/pixels, barrel/disk/cone) in same fit

Silicon Detectors — SI

Preparations for track reconstruction:

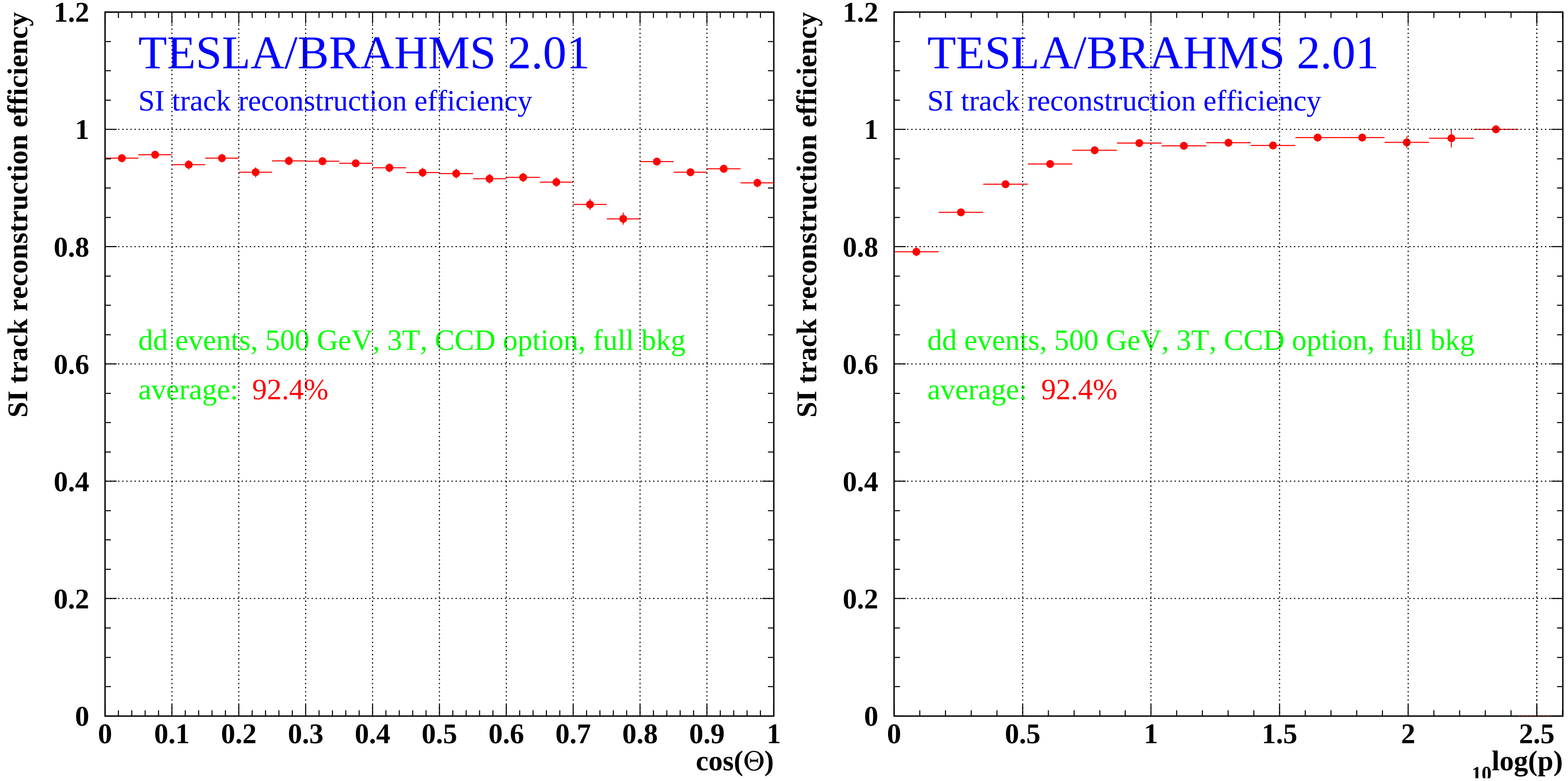
- all possible hits on strip detectors are collected
  (mark mirror hits as mutually excluded, but keep them all)
- translate cone and disk hits into pseudo-barrel hits
  (r position assumed exact, translate r error into z error for given Θ)

$\longrightarrow$ only "barrel" "pixels" in fit!

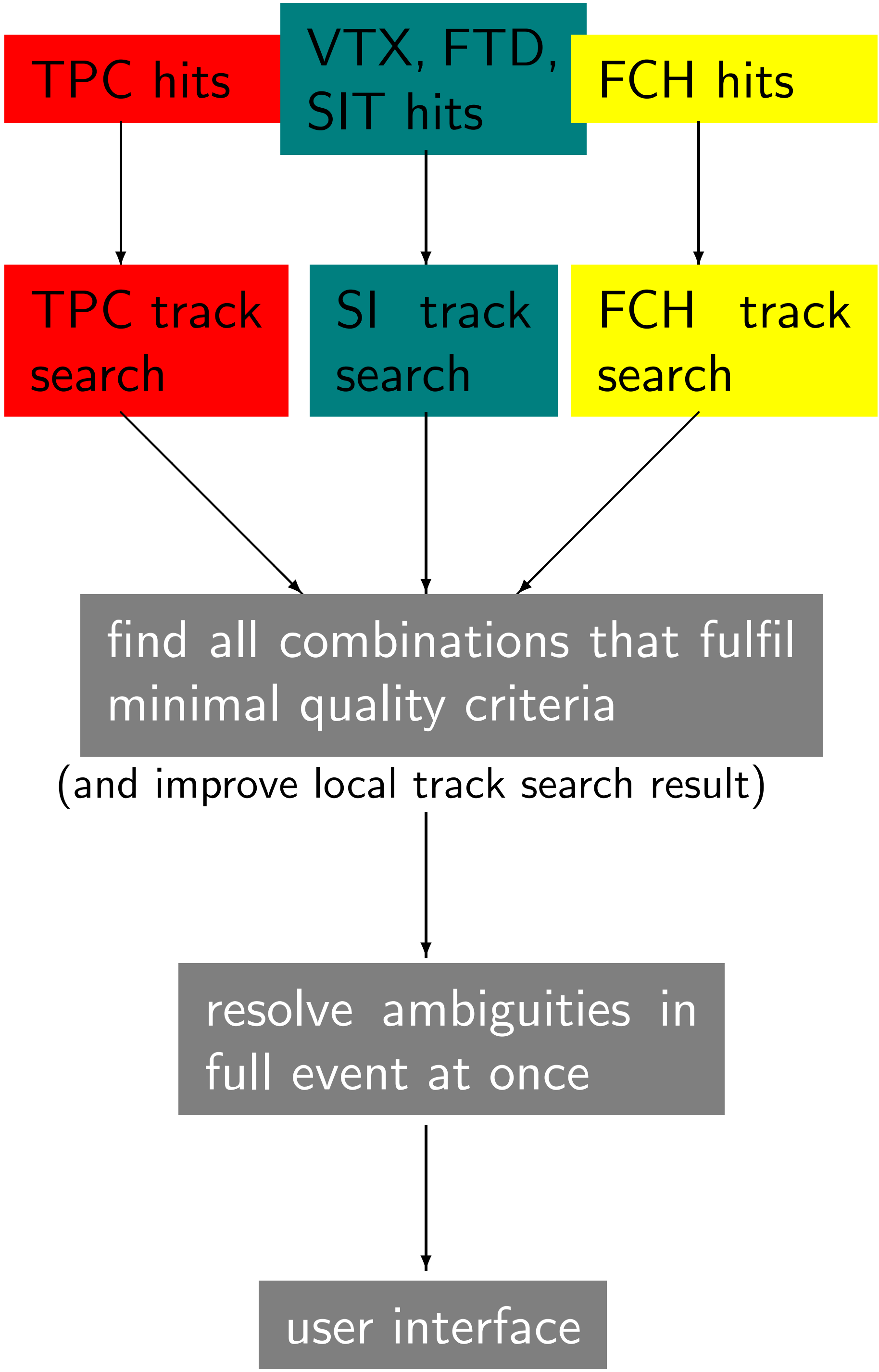
Track reconstruction algorithm:

- sort hits into φ and Θ sectors (overlapping $\longrightarrow$ no boundary effects)
- do track search only in sectors with enough layers hit
- search 3 hit seeds on outer layers that pass helix fit (χ^2, d_0, z_0 cuts)
- add hits on inner layers that survive helix fit with seed
  (other than TPC: try fit after each step!)
- keep all surviving tracks, even if same hit used in multiple candidates
- combine results of all φ and Θ sectors
- final fit (incl. energy loss)

Silicon Detectors — SI



plots for TESLA CCD with 60BX noise (60,000 noise hits on inner layer!)
problems in overlap between barrel and endcap (left)
problems for tracks with significant curvature (right)
but: global track search recovers almost entire inefficiency!



Global Track Search

1. local track search with Kalman filters based on software from ALEPH, OPAL

2. connect pieces to ambiguous track candidates using DELPHI software

3. optimize assignment of hits to tracks on global event basis (DELPHI ambiguity resolver)

Global Track Search

very flexible system: perform arbitrary predefined global track searches

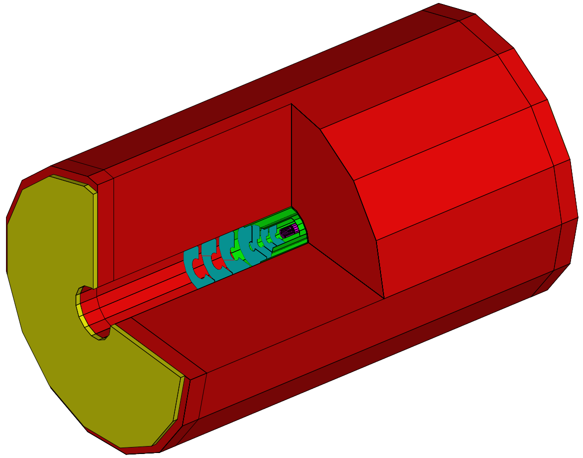
every track search:

- start with all local tracks in one detector (e.g. TPC)
- do helix extrapolation to other detectors (e.g. SI, FCH)
- for each matching combination: attempt full fit

full fit can remove individual subdetectors/SI layers to improve fit
(unless vetoed by user — e.g. usually don't remove full TPC track)

bad detector combinations can be excluded
(e.g. TPC + inner VTX layer only)

Global Track Search



BRAHMS 2.01:

1. SI tracks $\longrightarrow$ TPC tracks, FCH tracks
2. TPC tracks $\longrightarrow$ SI tracks, FCH tracks
3. combined tracks found in 1–2 $\longrightarrow$ TPC tracks, FCH tracks, SIT hits
4. combined tracks found in 1–3 $\longrightarrow$ FTD hits, VTX layer 3–5 hits
5. combined tracks found in 1–4 $\longrightarrow$ VTX layer 2
6. combined tracks found in 1–5 $\longrightarrow$ VTX layer 1

note: SI hits used not only as tracks from standalone SI patrec,
but also as separate hits (incl. hits not assigned to SI tracks).
SI hits removed by full fit at early stage may be reassigned later

idea of this specific track search strategy:
most precise extrapolation to inner VTX layers using almost full tracks
$\longrightarrow$ best possibility to deal with huge background

Global Track Search

track candidates at this stage are ambiguous!

- might share hits or track parts (e.g. same FCH track)
- might consist of mutually excluded (mirror) hits

$\longrightarrow$ taken care of in the final reconstruction stage

recursive algorithm (C code):

- divide set of tracks into disjunct subsets:
  all tracks of each subset connected by exclusions,
  no track of any subset excluded against any track of *other* subset
- resolve ambiguities within subsets by
  removing mirror hits
  removing multiply used hits from all but one track

which hits are to be removed where?
— optimal solution maximizes score function

Global Track Search

score function

weighted sum of

- total number of tracks
- logarithm of fit χ^2 probability
- bonus for each subdetector used (different value for each detector!)

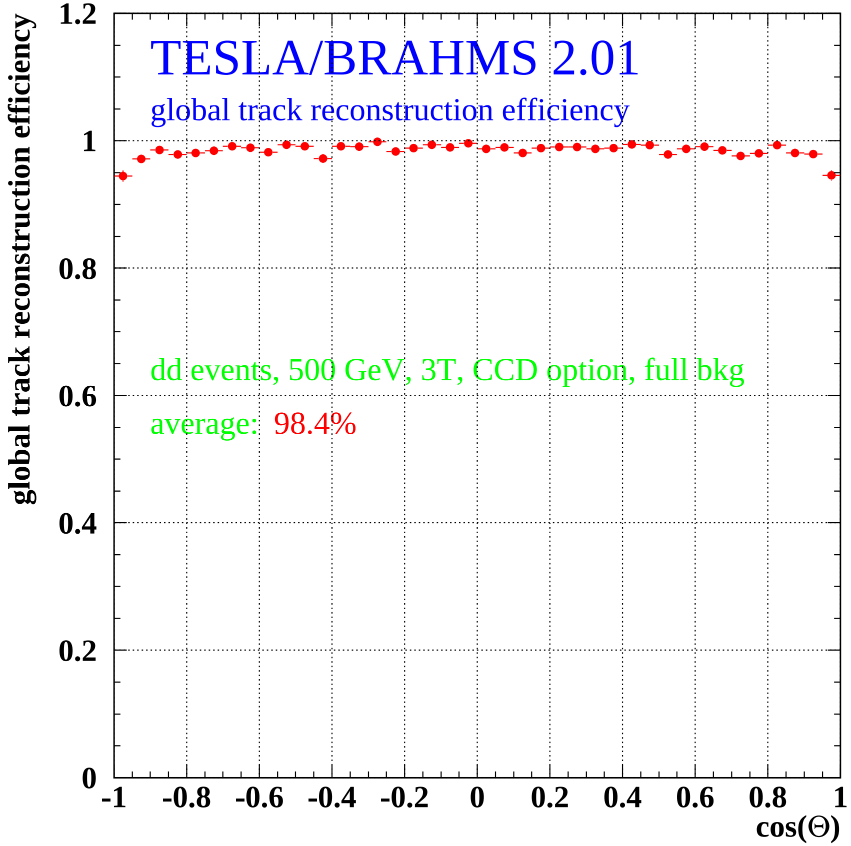
note:

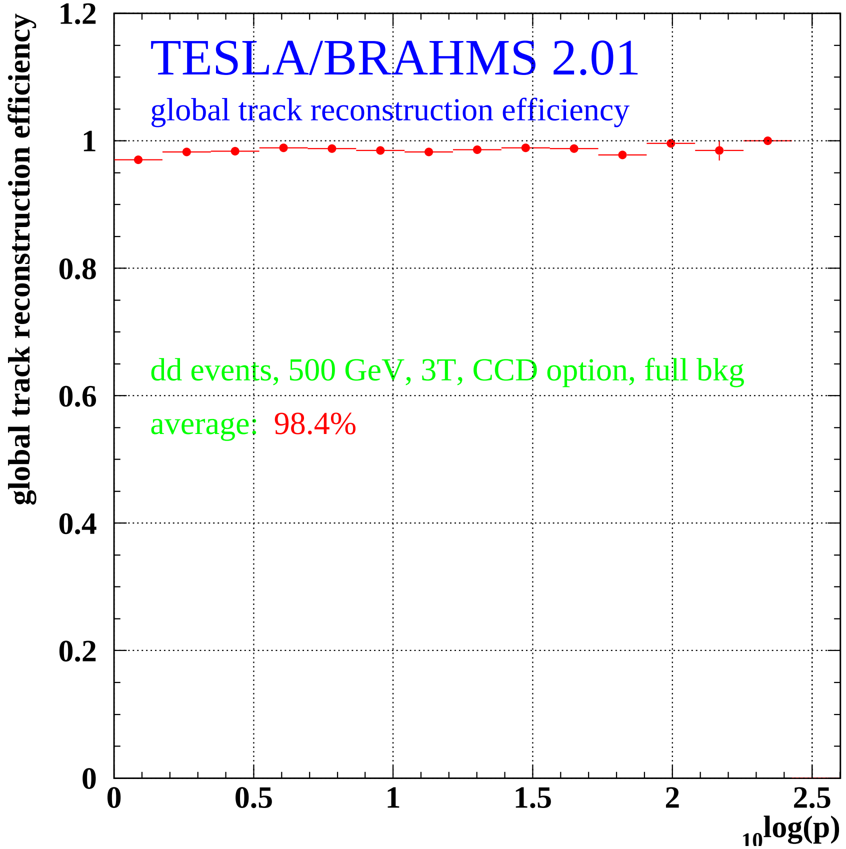
recursive algorithm very slow in especially complicated events

$\longrightarrow$ timeout

$\longrightarrow$ fallback algorithm: cut ambigous hits away from tracks with worst χ^2

Global Track Search





good results: inefficiency $< 2\%$, fake rate $< 1\%$, split tracks $< 5\%$

Summary

BRAHMS 2.x/3.x tracking

- is flexible: new detector setups and search strategies easy to implement
- has many bells and whistles: many parameters waiting to be optimized
- is hardly maintainable: $\mathcal{O}(10\text{–}100)$ authors, partially messy
- is *way* to slow for most useful applications